ANT - DATA TYPES

http://www.tutorialspoint.com/ant/ant_data_types.htm

Ant provides a number of predefined data types. Do not confuse the data types that are available in the programming language, but instead consider the data types as set of services that are built into the product already

The following is a list of data types provided by Apache Ant

File Set

The Fileset data types represents a collection of files. The Fileset data type is usually used as a filter to include and exclude files that match a particular pattern.

For example:

```
<fileset dir="${src}" casesensitive="yes">
  <include name="**/*.java"/>
  <exclude name="**/*Stub*"/>
</fileset>
```

The **src** attribute in the above example points to the source folder of the project.

In the above example, the fileset selects all java files in the source folder except those that contain the word 'Stub' in them. The casesensitive filter is applied to the fileset which means that a file with the name **Samplestub.java** will not be excluded from the fileset

Pattern Set

A pattern set is a pattern that allows to easily filter files or folders based on certain patterns. Patterns can be created using the following meta characters.

- **?** - Matches one character only
- ***** - Matches zero or many characters
- ****** - Matches zero or many directories recursively

The following example should give an idea of the usage of a pattern set.

```
<patternset >
  <include name="src/**/*.java"/>
  <exclude name="src/**/*Stub*"/>
</fileset>
```

The patternset can then be reused with a fileset as follows:

```
<fileset dir="${src}" casesensitive="yes">
  <patternset ref/>
 </fileset>
```

File List

The File list data type is similar to the file set except that the File List contains explicitly named lists of files and do not support wild cards

Another major difference between the file list and the file set data type is that the file list data type can be applied for files that may or may not exist yet.

Following is an example of the File list data type

```
<filelist >
  <file name="applicationConfig.xml"/>
  <file name="faces-config.xml"/>
  <file name="web.xml"/>
  <file name="portlet.xml"/>
</filelist>
```

The **webapp.src.folder** attribute in the above example points to the web application's source folder of the project.

Filter Set

Using a Filter Set data type with the copy task, you can replace certain text in all files that match the pattern with a replacement value.

A common example is to append the version number to the release notes file, as shown in the example below

```
<copy todir="${output.dir}">
  <fileset dir="${releasenotes.dir}" includes="**/*.txt"/>
  <filterset>
    <filter token="VERSION" value="${current.version}"/>
  </filterset>
</copy>
```

The **output.dir** attribute in the above example points to the output folder of the project.

The **releasenotes.dir** attribute in the above example points to the release notes folder of the project.

The **current.version** attribute in the above example points to the current version folder of the project.

The copy task, as the name suggests is used to copy files from one location to another.

Path

The **path** data type is commonly used to represent a classpath. Entries in the path are separated using a semicolon or colon. However, these characters are replaced a the run time by the running system's path separator character.

Most commonly, the classpath is set to the list of jar files and classes in the project, as shown in the example below:

```
<path >
  <pathelement path="${env.J2EE_HOME}/${j2ee.jar}"/>
  <fileset dir="lib">
      <include name="**/*.jar"/>
  </fileset>
</path>
```

The **env.J2EE_HOME** attribute in the above example points to the environment variable **J2EE_HOME**.

The **j2ee.jar** attribute in the above example points to the name of the J2EE jar file in the J2EE base folder.